TARGET-DRIVEN PERCEPTUAL GROUPING

IAN JERMYN, DAVID JACOBS, AND DAVI GEIGER

Courant Institute, New York University, New York 10012

NECI, 4 Independence Way, Princeton, NJ 08540

Courant Institute, New York University, New York 10012

Abstract. We present an approach to the perceptual organization of closed contours into parts that integrates both bottom-up *and* top-down information into the grouping process. The top-down information is given in the form of a "target", a contour with a pre-computed part structure. Our model gives us the ability to deal with image contour distortions that would render purely bottom-up procedures unsuccessful. It also allows different hierarchical levels in the description of an object to influence perceptual grouping. As output we find not only the part structure of the image contour, but also a correspondence to the target. Moreover, a naturally asymmetric measure of similarity is obtained, as suggested by psychophysical data.

We represent the part structure of a contour as a self-matching, a pairing of the contour with itself. It incorporates region and symmetry information into a tree structure that describes the contour parts and their relations. To compute the self-matching for the image contour we minimize an energy functional that globally couples the image to its self-matching using local geometric features, and couples the image self-matching to the part structure of the target. In this way different levels of description influence grouping. We minimize the energy using variants of Dijkstra's shortest paths algorithm, a feature that is of interest in its own right.

Extensions of the work to grey-scale images and variable targets are also discussed.

1. Introduction

A fundamental problem in vision understanding can be illustrated by a famous example, figure 1. The difficulty of performing perceptual grouping in this image is apparent and confounds even human observers, but the task becomes considerably more manageable when the observer is informed that the image contains a dog. The pieces then come together and grouping is relatively easy. Clearly bottom-up clues and top-down knowledge of the target of the grouping are interacting and making organization possible. The same type of interaction occurs throughout vision, and especially in perceptual organization, where different hierarchical levels of image interpretation are linked. In this paper we study this interaction using a simplified domain, and describe and implement a model of the process in that domain.

jermyn@cs.nyu.edu, dwj@research.nj.nec.com, geiger@cs.nyu.edu.

We use images consisting of closed contours, and try to move to a higher-level representation by grouping the interior into parts. This problem has been much studied and is the principle subject of, for example [2, 3, 14, 17, 21, 22, 27]. Much of this work is aimed at understanding shape, because of the fundamental role that this seems to play in object recognition: psychological work, for example [20], suggests that basic- or entry-level classification is highly correlated with shape. Reaching such an understanding is a difficult enough task that it is worth investigating within the context of a simplified domain. The domain of closed contours serves the purpose well, preserving the essentials of shape while ignoring the complexities of grey-scale images. It is also practically relevant if contours are manually extracted from grey-scale images (using "snakes" for example, [13]) prior to the type of processing described here. Human beings find it easy to recognize shapes from contours and this renders them interesting from a cognitive point of view as well. Other related work includes [4, 5, 6, 8, 10, 11, 12, 18, 19, 24, 26].

The grouping of contours into parts is also the subject of [9], (hereafter referred to as LGK) and it is on this that the present work is built. They use a model of non-rigid contour self-matching that allows the derivation of a part structure for a contour that has with much in common with the symmetry axis. The use of contour matching in recognition is investigated in, for example [1, 8, 7, 23]. A non-rigid correspondence is set up between the points in one contour and those in another, typically using an energy functional to control the correspondence. In LGK, a contour is matched to itself, in an orientation reversing fashion, in order to capture region and symmetry properties of the contour. The interior of the contour is divided into parts, represented by edges, while part junctions are represented by nodes (figure 2).

Like LGK, we use an energy functional to define what we mean by a good grouping. We use a similar bottom-up energy to theirs to guide the construction of a self-matching from the contour, but in our case the emerging part structure is also compared to a target consisting of a contour and its pre-computed self-matching, and this is used in addition to the bottom-up energy to direct the grouping. To be specific, we minimize the sum of the energies associated with the self-matching and the comparison to the target, thereby coupling the higher-level representation produced by grouping to both the target and the image. To solve the model we use an improved algorithm, a variant of Dijkstra's shortest paths algorithm, a feature that we believe is of interest in its own right. The outcome is the perceptual grouping of the contour into parts that best matches the target while still being consistent with the image. In addition, we find a correspondence to the target and a measure of how good that correspondence is, or in other words, recognition data.

The advantages of our approach can be divided into two groups: the practical and the conceptual. On the practical level, it allows us to deal with distortions and variations in the image that would render pure bottom-up extraction of a part structure, and subsequent recognition, unsuccessful (for example, figure 5).

By comparing to the target at the level of parts it also ameliorates the problems associated with using non-rigid contour matching to compare shapes, as is done in [1, 8]. Not only are the combinatorics frightful in that case, but regions and symmetries are ignored, with the result that perceptually quite distinct distortions of a contour may be treated equally [1, 8]. On a conceptual level, comparing to the target at the level of parts rather than at the level of contour points introduces a representation hierarchy, and shows one way in which more abstract knowledge can affect the grouping process. We also believe that entwining perceptual grouping and recognition in one procedure is the right way to deal with the problems encountered by purely bottom-up approaches, where the significance of image features can be highly ambiguous.

In the next section we describe the energy functional that we use to define the self-matching and hence the part structure of contours. We follow this in section 3 with a discussion of the algorithm used to minimize this energy. Then, in section 3.3 we present some results from our implementation of this algorithm. Finally we conclude with a summary and a discussion of future work. An appendix describes the general framework within which we work and gives some mathematical details omitted from the body of the paper.

2. The Model

We begin by defining a self-matching, as this is one of our central concepts. It is expanded upon further in A.2. A self-matching ρ establishes a pair-wise correspondence between the points of a contour that obeys certain continuity restrictions. It should be orientation-reversing and piece-wise differentiable. It should also obey a reflection symmetry that ensures that when $\rho(x) = y$, $\rho(y) = x$. Thus every point on the contour has a partner, which divides the interior region into infinitesimal strips joining the partner points, except at discontinuities, where a point can be regarded as having two or more partners. We enforce the twin rules that a point cannot have more than two partners, and that if point x has y and z as partners at a discontinuity, then y must have z as a partner. Drawing the midpoints of the lines joining the partner points then leads to a degree three tree structure similar to the symmetry axis (figure 2)[1]. The leaf nodes of the tree structure lie on the contour itself, and are the points whose partners are themselves ($\rho(x) = x$). The internal nodes correspond to groups of three discontinuities in the self-matching, as enforced by the rule. We will refer to this discrete tree structure as the self-matching also and use the same symbol for it as for the full map.

A self-matching describes the decomposition of the object into parts. The regions defined by the edges are the parts and the nodes are either part junctions, or terminations of parts. Of course for this to be a useful description, we have to be able to define which of the infinity of possible self-matchings of a contour Γ actually represents its part structure. Consistent with our desire to use top-down information to perform

[1]The rule disallows degree four or higher trees. The justification for this is that such points are possible but are structurally unstable.

this grouping, we will begin not only with a contour, but with a "target", another contour with a pre-defined part structure in the form of its own self-matching $\rho_{\mathcal{C}}$. The self-matching of the target will be defined in a purely bottom-up way using part of the model we present here. It represents the concept for which we are looking, in the language of the appendix. We will define an energy functional below, whose minimum will give us both the self-matching of Γ, $\rho_{\mathcal{I}}$, and a correspondence between $\rho_{\mathcal{I}}$ and $\rho_{\mathcal{C}}$, μ, that will tell us the part structure of Γ in terms of the target. It will also give us a measure of the similarity between the image, viewed as an instance of the target, and the target.

Thus we must define an energy E that is a functional of Γ, $\rho_{\mathcal{I}}$, $\rho_{\mathcal{C}}$ and μ. We will restrict the form of this energy functional to the following two terms:

$$E(\Gamma, \rho_{\mathcal{I}}, \rho_{\mathcal{C}}, \mu) = E_{\mathcal{I}}(\Gamma, \rho_{\mathcal{I}}) + E_{\mathcal{C}}(\rho_{\mathcal{I}}, \rho_{\mathcal{C}}, \mu) \tag{2.1}$$

There are alternatives to this form of the energy function, and some are discussed in section A.1, but the independence assumptions it incorporates (if thought of as log probabilities) are reasonable. They state that the representation of the image as a self-matching depends on the representation of the target and the image, but that the correspondence between the two representations is independent of the original image except as summarized in its representation. The decomposition therefore introduces a hierarchy in the description of the object.

The first term, $E_{\mathcal{I}}$, is similar to the model used in LGK. We will describe it briefly here: further details can be found in appendix A.2.1 and the original paper. The second term, $E_{\mathcal{C}}$, describes the correspondence between the image self-matching and the model self-matching, and incorporates the top-down information that we wish to use to guide the derivation of $\rho_{\mathcal{I}}$. Bottom-up approaches would minimize $E(\Gamma, \rho_{\mathcal{I}})$ first to find $\rho_{\mathcal{I}}$, followed by a minimization of $E(\rho_{\mathcal{I}}, \rho_{\mathcal{C}}, \mu)$ to find μ. This procedure decouples $\rho_{\mathcal{I}}$ from $\rho_{\mathcal{C}}$. By minimizing the sum of the terms simultaneously we couple the two variables and allow top-down information to influence bottom-up procedures.

2.1. $E_{\mathcal{I}}$**: bottom-up cues.** The energy functional $E_{\mathcal{I}}$ is defined by integrating a density along the contour. At each point x of the contour at which the self-matching is continuous, the density measures two things: how close the tangents at x and its partner are to being co-circular (or mirror-symmetric), and how much the speed of motion along the contour differs for x and $\rho(x)$ (the "stretching"). If the stretching is unity and the tangents are co-circular the density contributes zero. Thus it is easy to see for example that any self-matching of a circle defined by reflection in a diameter will have energy zero. At points of discontinuity the density becomes a delta function, thus contributing a cost associated to the node of the tree structure. This was taken to be a constant in LGK. We take it to be equal to a constant times the perimeter of the triangle formed by the lines joining the three matched points. This favours treating as parts regions that are more

obviously "cut off" than others, but does not otherwise seem to affect the results greatly. This completes the summary of the term $E_{\mathcal{I}}$. A technical discussion and definitions can be found in appendix A.2.1.

2.2. $E_{\mathcal{C}}$**: top-down information.** To describe the second term in the energy, $E_{\mathcal{C}}$, we have to define what we mean by a correspondence μ of the self-matching of the image $\rho_{\mathcal{I}}$ to that of the target $\rho_{\mathcal{C}}$. We want this comparison to use the discrete tree structure of the self-matchings for two reasons. The first is that we are trying to describe how higher levels of description, in this case in terms of part structure, can influence perceptual grouping. This is one of the motivations for using the particular representation we have chosen in the first place. The self-matching incorporates important structural information about the object and we should use this. Second, as discussed in the introduction, pure top-down approaches that use non-rigid matching between shapes suffer from a number of defects: computational complexity and unintuitive measures of similarity among them [1, 8]. Although algorithmic efficiency is not our main goal, we wish to test the idea that using bottom-up cues to construct parts can reduce the complexity of the top-down process and hence allow that information to be used more effectively.

With these motivations in mind, we define an isomorphism μ between two self-matchings as follows[2]. The first component is an isomorphism between the tree structures of the self-matchings, or in other words a one-to-one correspondence between the part structures of the image and target. The second component is a set of bijections between the corresponding edges of the tree structures. This establishes a precise map between corresponding parts.

We now have to define an energy on the space of self-matching isomorphisms (see also appendix A.3.1). Following our motivation, this energy will take the form of a sum over the part structure, which leaves open the choice of an energy to express part similarity. One possibility is to define an energy analogous to $E_{\mathcal{I}}$, that looks at the details of any mapping between the parts to see how well they compare. The problem with this is that the space of self-matching isomorphisms is huge. In fact it is exactly this type of comparison that we mentioned in connection with top-down contour matching approaches, and that we wish to avoid. It also goes against the spirit of the part comparison. Instead we use the equivalence class structure given by the tree isomorphisms and assign an energy that is constant on each class. This is equivalent to reducing the size of the space of mappings between self-matchings to that of the space of tree isomorphisms. We simply pick a distinguished member of each equivalence class and use that to define the energy. The choice we make is to map the parts linearly into one another along the edges of the tree structure. Using this choice of map between corresponding parts, we have defined two different measures of part similarity. The first simply compares the lengths of the two parts. This works well (in principle and practice) in cases where the

[2]We will assign infinite energy to any correspondence between self-matchings that is not an isomorphism in the sense defined here, although we do not in principal discount the utility of more general morphisms.

target has a well-defined articulated part structure, and the image, while in general similar to the target, has additional parts that are not well-articulated or parts that are shrunk relative to the target in some way. This will help with occlusions of certain types, pose changes to some degree, and additional objects in the image that contribute noise to the contour (if we imagine it has been extracted from a grey-scale image). The second measure is a little more detailed. The energy functional sums contributions computed by comparing infinitesimal quadrilaterals formed in both image and target by the tangent vectors of matched points along the tree edges. Two quadrilaterals are compared by summing energies computed from the eigenvalues of transformations needed to map one into the other. This is designed to compare the shapes of the two parts in a more precise way. To some extent this establishes a recursive hierarchy, since our original problem is closely connected to shape comparison. The intention is that individual parts will have a somewhat trivial shape and that simple measures will capture this adequately.

We thus have our mathematical model of grouping using top-down information. We are left with the problem of how to solve it: given Γ and $\rho_{\mathcal{C}}$, how do we find the values of $\rho_{\mathcal{I}}$ and μ that minimize the energy 2.1? The answer to this lies in the next section.

3. The Algorithm

In the following subsection we describe the variant of Dijkstra's shortest path algorithm that we use to find the minimum of the first term in equation 2.1. This is an improvement both conceptually and in complexity to the algorithm described in LGK. There the complexity forced a drastic sub-sampling of the contour ("special points") to reach reasonable execution times. In fact this technique worked quite well and is also open to the current algorithm. However, considerations connected to grey-scale images pushed us in the direction of using all the contour points for the computations and the new algorithms render this possible. It is unclear which technique will prove more useful in the long run. Partly this depends on whether better techniques can be found for identifying qualitatively important points on the contour given that the target is known. Conceptually the algorithm is an improvement because it acknowledges the topology of the closed contour, and does not choose a arbitrary marked point to reduce the problem to the simply connected case. Then in section 3.2 we describe the alterations to the algorithm that enable it to find the minimum of the combined bottom-up and top-down energies, E in equation 2.1. Throughout we assume that the contour has been sampled and that we are dealing with the discrete set of sampled points, while still having access to the ordering of the points implicit in the contour. We use the term "node" to refer to the vertices in the self-matching tree, and the term "vertex" to refer to the vertices in the discretized graph of the self-matching.

3.1. Algorithm for $E_{\mathcal{I}}$.

As mentioned, the algorithm that we use to minimize the bottom-up part of the energy functional in absence of a target is essentially Dijkstra's algorithm for shortest paths in a graph. To

get an immediate intuition as to why this works, imagine trying to find the continuous orientation-preserving map from the unit interval to itself that minimizes the integral of some density along the interval. Drawing the graph of the map in the unit square and then discretizing the x and y axes makes it clear that at the discrete level what is sought is a path in the lattice, from the point $\langle 0,0\rangle$ to the point $\langle 1,1\rangle$, of minimum total energy. The edges in the path are not restricted to the rectangular lattice edges only however, but can be from any $\langle x,y\rangle$ to $\langle x+\delta,y'\rangle$ where δ is the x lattice spacing and y' is any value on the lattice between 0 and 1. If the density has no second derivatives or higher, the energy of the whole path is the sum of incremental energies due to each edge. By defining the edges as above, in the direction of increasing x only (or in the language to be used below, by restricting the "moves" to be in the direction of increasing x) and running Dijkstra's algorithm with $\langle 0,0\rangle$ as a source, we can check on each iteration whether a path to $\langle 1,1\rangle$ has been found. When it is found, it must be the lowest energy such path because of the workings of the algorithm. The restricted direction of the "moves" ensures that it has covered all x and y values and is thus complete. In this way we find the global solution to the minimization problem.

In our case the algorithm has to be altered somewhat, for three reasons. First, we are no longer dealing with the unit interval, but with closed contours, or effectively with S^1. The extra topology means that we can start from any point on the diagonal set, $\{\langle x,x\rangle : x\in\Gamma\}$: there is no preferred starting point as there is no boundary to the circle, unlike the interval. This has the consequence that it is harder to define a notion of direction; nevertheless it is necessary, as in the simply connected case, to ensure that the resulting map is "complete" or a bijection. It will turn out that "direction" is path dependent. Second, we must impose the symmetry condition as we are dealing with a *self*-matching. This means that half of the data is irrelevant. Once we know that x maps to y, we know what y maps to. Third, we are allowing discontinuities in our maps. This means we must allow two ways of extending any path, two different types of edges or moves, corresponding to continuous or discontinuous progress, and along with this, ways of imposing the rule that limits the self-matching to be a tree of degree three. The first two reasons mean that the graph of the self-matching will be drawn in $S^1\times S^1/\mathbb{Z}_2$, the space of unordered points on the two-torus, as opposed to $\mathbb{R}\times\mathbb{R}$ in the example in the first paragraph. This space has a boundary, the diagonal set mentioned above. We will refer to points in this set as "boundary" points, or "boundary" vertices if we are dealing with the discretized case. Points in $S^1\times S^1/\mathbb{Z}_2$ at which there is a discontinuity will similarly be referred to as discontinuity points or vertices. The third reason leads to operational differences in the algorithm itself. There are indeed two different types of moves, and vertices are allowed to have two predecessors (but no more) to enforce the tree structure. To picture the workings of the algorithm in this more involved setting, visualize the growth of the tree. Beginning from every point on the contour, which is isomorphic to the boundary set, the tree grows outwards. At any point in time the growth that leads to the lowest overall

cost is made, as per Dijkstra, until one of the tendrils reaches the "other side" of the contour and completes itself. If a discontinuity is the cheapest way to move forward then the algorithm checks that an appropriate partner path has already been investigated and joins with it before continuing on.

To describe the algorithm it is therefore necessary to describe how we deal with these complications. The possible moves that can be made from a given vertex will depend not only on the vertex but on its predecessor(s) (the path dependence mentioned above), and as promised will be of two types, corresponding to continuous and discontinuous growth of the map. The dependence on predecessors is to ensure that a consistent direction is maintained for the growth of the path. In the picture at the end of the last paragraph it is easy to see that we must prevent the tree structure from doubling back on itself. Since a given vertex (pair of matched points) divides the interior of the contour into two parts via the line joining the points, it may be approached from two directions. It can only move forwards in a direction opposite that of its predecessors. The possible moves forward from a given vertex are described in figure 3. It is important to notice that a discontinuous move can only be made when it can merge with an appropriate second path that has a known energy, which is to say that its vertices have all been removed from the queue. In the absence of such a path the move is not allowed and must wait until such a path does exist.

Given this information Dijkstra'a shortest path algorithm proceeds as usual. The elements of the initial set are the boundary vertices with zero energy. Second copies of these exist in the queue but with infinite energy, corresponding to terminal points. On each iteration, the vertices corresponding to the possible moves from the vertex that has been removed from the queue are updated, using both continuous and discontinuous moves. The algorithm proceeds until a boundary vertex is pulled off the queue. The restriction of the direction of moves ensures that when such a vertex is encountered, the whole contour has been self-matched. Note that the nodes of the self-matching tree structure resulting from the graph of the self-matching produced by the algorithm are either boundary vertices (leaf nodes of the tree structure) or groups of three discontinuity vertices (internal nodes of the tree structure). The edges of the self-matching tree structure are formed by the other vertices in the output of the algorithm.

3.2. **Incorporating the target.** We will incorporate the target by allowing the comparison between the part structures of the image and target, as described by their self-matchings, to influence the search carried out in the algorithm described above. At the same time we will build the tree isomorphism that relates the image self-matching to the target self-matching. This isomorphism means that we need a correspondence between the boundary and discontinuity vertices of the output of the algorithm and the leaf and internal nodes of the self-matching of the target, as described at the end of the previous subsection.

We will expand the algorithm of the previous section by allowing vertices to be of two classes: those that are not and those that are mapped to a corresponding target node. Those in the second class also record to

which node they are mapped. From the previous paragraph it is clear that only boundary and discontinuity vertices can ever be in the second class. An invariant will be maintained: whenever a vertex is removed from the queue, all the boundary and discontinuity vertices among its predecessors lie in the second class. When a boundary or discontinuity vertex v is removed from the queue, it is mapped to a node in the target in all ways geometrically consistent with the mappings of the boundary and discontinuity vertices among its predecessors. For each one of these mappings a vertex is created in the second class and placed in the queue. This vertex has the same predecessors as v, but its energy is incremented from that of v by the cost of mapping to the target the parts or edges between v and its nearest boundary or discontinuity predecessors. Its energy is therefore equal to the sum of the cost of the self-matching represented by its predecessors, and the cost of the partial tree isomorphism to the target represented by the mappings of itself and its boundary and discontinuity predecessors. Figure 4 offers a picture of the process.

There is one subtlety that arises from efficiency issues. If the invariant were enforced from the beginning of the algorithm it would imply that we would have to map every initial boundary vertex in every possible way to the target and create vertices in the second class for all these mappings. This is simply not efficient. Instead, when a discontinuity vertex is removed from the queue, any boundary predecessor vertices are matched together with the discontinuity vertex.

A solution is found when a boundary vertex in the second class is removed from the queue. The workings of the algorithm ensure that this represents a complete self-matching, because the moves that a path can make to extend itself have not changed, and that all the boundary and discontinuity vertices among its predecessors belong to the second class. It is thus a self-matching and a correspondence to the target. Its energy is the evaluation of equation2.1 on this self-matching and correspondence.

Using the target buys other computational advantages. It renders the algorithm, already quite stable to parameter changes, stable over a much wider range of parameter values. This is as it should be, since we can view changes in parameter values as just another type of noise, and part of the motivation for using a target was to render the computation more stable. This in turn implies that a learning procedure to tune the parameter values should converge much more rapidly than it would without the target.

3.3. **Examples.** Figures 5, 6, and 7 show some examples of results obtained from the algorithm. They all contrast self-matchings (and hence part structures) found with, (b), and without, (a), the target, using the algorithms of section 3.2 and section 3.1. They thus compare the effects of using $E_{\mathcal{I}}$ alone, or the combined energy E.

4. Conclusion and Future Work

The work in this paper can be seen from a number of perspectives that help clarify its aims. The problems of pure bottom-up perceptual organization are well-known and we have tried to address them using a particular example: grouping the elements of a closed-contour representation of a shape into parts. This is achieved through two mechanisms. One is the use of a stable representation of part structure that is fairly immune to noise in the contour to begin with. This alone however cannot deal with the central problem, which is that the task of perceptual grouping (particularly into parts) may not be well-defined without top-down knowledge of what the object represented actually, or at least might be. It is in addressing this problem that the main thrust of the present work is located. We have described a mathematical model and an algorithmic implementation of a mechanism whereby top-down knowledge can influence the perceptual grouping outcome. As it stands this mechanism enables the construction of the correct part structures for contours that would normally be misinterpreted by processes that use bottom-up information only. As mentioned in the text, the work can also be viewed as an attempt to address the computational complexity problems normally associated with some pure top-down processes. A third point of view is to place the work within the context of more general object recognition, which appendix A.1 attempts to make more precise. This point of view arises because we do not simply construct a representation of the image contour that depends on the target, but also construct an explicit correspondence to the target and a measure of the similarity between it and the image. This measure of similarity is asymmetric, which is consistent with psychophysical work noted in [15] and described in [25, 16]. The question that we are asking is altered slightly. Instead of "does it look like a dog?", the question is now "if I think of it as a dog, does it look like a dog?" We believe that this is the correct form of the question at almost all levels of vision except perhaps the very low-level.

There are many possible extensions to the current work, each of which emphasizes a different aspect. One important direction is to move towards grey-scale images. This can be achieved in a few distinct steps. The first is to abandon the explicit contour representation used at present, and deal with occluded contours and irrelevant data. This is of interest in itself, as it may allow comparison with human data for object search in a background of distracters. The second is to move to black and white images. The third is to move to full grey-scale images. These steps are currently under way. We expect even more significant advantages to our approach to become apparent with the extension to grey-scale images. The comparison of image to contour, contour to part structure and part structure to target should make possible the extraction of objects such as the dalmatian in figure 1.

Another extension is to emphasize the recognition aspects of the problem and create a more variable representation of the target (called "concept" in the appendix). Examples would be allowing variation in

articulation or pose. This would render the process of matching to the target more stable and would also allow discovery of information about the object being viewed from the particular target variation chosen.

There are also technical improvements to be made. Work is in progress on a different species of algorithm for the minimization. It also seems important to emphasize the qualitative nature of the picture the self-matchings summarize, for reasons of both efficiency and principle.

Appendix A. Appendix

A.1. **Framework.** We describe an abstract model of recognition within which to fit the perceptual grouping work.

We start with a space of images $\mathcal{I}$ and a space of concepts $\mathcal{C}$. The latter is clearly not well-defined but can be thought of as a set of basic-level descriptions: dog, cat, chair etc. We use it to describe the model but never use it directly.

We wish to allow the possibility of representations of both concepts and images, which are then used in place of the originals for recognition purposes. To model this we create two bundles, the representation spaces $\mathcal{R}_\mathcal{I} \to \mathcal{I}$ and $\mathcal{R}_\mathcal{C} \to \mathcal{C}$ over image space and concept space respectively. Objects in the two representation spaces, while not in the same category, may be mapped by forgetful functors into a single category and hence there is a functor to the set of morphisms between the elements of any point $\langle r_\mathcal{I}, r_\mathcal{C} \rangle \in \mathcal{R}_\mathcal{I} \times \mathcal{R}_\mathcal{C}$. This allows several ways of matching an image representation to a concept representation, and creates another bundle: the matching space $\mathcal{M} \to \mathcal{R}_\mathcal{I} \times \mathcal{R}_\mathcal{C}$.

Given an image $i \in \mathcal{I}$ and a concept (or set of concepts) $c \in \mathcal{C}$, recognition consists in finding a point $m \in \mathcal{M}$ over i and c. There are many procedures for finding such a point and it is these that distinguish the many models of recognition. In typical "bottom-up" models a point is first found in $\mathcal{R}_\mathcal{I}$ that depends only on the image i, which is to say that a representation of the image is found. The important point is that the representation does not depend in any way on the concept for which we are searching. Once a representation has been found it may be compared to representations of concepts to identify the object.

We base our approach to finding the point in $\mathcal{M}$ on the minimization of an energy function $E(m)$. In principle this energy function could depend arbitrarily on the point in $\mathcal{M}$. The meaning of this can be seen by breaking a coordinate system in $\mathcal{M}$ down into base and fibre coordinates, and doing the same for the representation spaces. Then a point in $\mathcal{M}$ will be labelled by a triple: $\langle \mu, \langle \rho_\mathcal{I}, i \rangle, \langle \rho_\mathcal{C}, c \rangle \rangle$. The coordinate μ labels the matching between the representations $\rho_\mathcal{I}$ and $\rho_\mathcal{C}$ of the image i and concept c. The energy function can vary arbitrarily with these coordinates.

In practice and in particular in our models this is not the case. The energy function is a sum of two terms that decouple some of the coordinates. In our models the energy function takes the following form:

$$E(\langle \mu, \langle \rho_{\mathcal{I}}, i \rangle, \langle \rho_{\mathcal{C}}, c \rangle \rangle) = E_{\mathcal{I}}(\rho_{\mathcal{I}}, i) + E_{\mathcal{C}}(\mu, \rho_{\mathcal{I}}, \rho_{\mathcal{C}}) \tag{A.1}$$

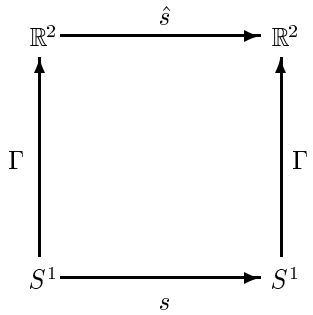
The first term describes the derivation of a representation from an image, whereas the second describes the derivation of the matching between representations of the concept and the image. Note that in the second term the image and concept do not appear explicitly, but they are present in the definition of the representations.

Of course there are many other ways to restrict the form of the energy function. Terms such as $E(\rho_{\mathcal{C}}, c)$ could be used to identify likely representations or poses, either due to an environment, or more practically, to allow variation due to pose and occlusions. The representation of the concept would then not need to be fixed in advance. In more general recognition tasks, where the concept is not identified a priori, a term like $E(c)$ can be used to express the likelihood of certain concepts being present in a given environment.

A.2. **Concept-independent model.** Here we describe details of the bottom-up model of section 2 and 2.1.

The image space is the space of all non-self-intersecting closed contours in the plane, or continuous piece-wise differentiable embeddings Γ of S^1 into $\mathbb{R}^2$, modulo diffeomorphisms of S^1, as described in the main text. While we describe our model the notation i for an image will be replaced by Γ, although strictly speaking it is $\mathrm{Im}(\Gamma)$ which is the image. We also replace the notation $\rho_{\mathcal{I}}$ for the representation of the image by $\hat{s}$ for the self-matching of Γ.

A self-matching is an orientation-reversing piece-wise differentiable bijection $\hat{s}$ from $\mathrm{Im}(\Gamma)$ to itself, with the property that $\hat{s}^2 = \mathrm{Id}$. The discontinuities in the self-matching will not be allowed to be arbitrary: they are limited by the rule described in section 2. Note that each self-matching $\hat{s}$ defines a map from S^1 to itself



via the following commutative diagram:

We will define the energy of the self-matching for each contour Γ using s in place of $\hat{s}$, pulling everything back to S^1.

The graph of a self-matching will be drawn in the space of unordered points on the two-torus, $S^1 \times S^1/\mathbb{Z}_2$, unordered because of the symmetry condition. This is an interesting topological object, being unorientable; it is a Klein bottle but with a hole cut in it, so that it has a boundary diffeomorphic to S^1—the boundary points in the algorithm in section 3.1. The graph has a natural tree structure, defined by the following sets of distinguished points: points on the boundary (representing contour points matched to themselves): these are leaf nodes; and those groups of three points ($\langle x, y\rangle$, $\langle y, z\rangle$, and $\langle z, x\rangle$) at which there are discontinuities: these may be thought of as grouped together into a single internal node for the purposes of the tree structure, and this is how we display them in the figures.

A.2.1. *Dynamics.* In this section we describe the energy term $E_{\mathcal{I}}$. We drop the subscript $\mathcal{I}$ where no confusion will be caused.

We are looking for an energy functional $E_{\mathcal{I}}(\hat{s}, \Gamma)$ on orientation-reversing piece-wise differentiable maps from the image of the contour to itself. We require this functional to be local; it will be the integral of an energy density $\hat{\mathcal{E}}(\hat{s}, \Gamma)$ over $\mathrm{Im}(\Gamma)$, re-expressed, as mentioned above, as the integral of a density $\mathcal{E}(s, \Gamma)$ over S^1. Using s as opposed to $\hat{s}$ simplifies the discussion but introduces issues of invariance that do not arise when we deal with $\hat{s}$ directly. There are two types of invariance issues: those associated with $\mathbb{R}^2$, and those associated with S^1 arising from the use of s. We want the functional to be invariant under global transformations of $\mathbb{R}^2$ to itself, to whit, translations, rotations and scaling[3]. Secondly, we want invariance under arbitrary reparametrizations of the contour, coordinate invariance on S^1, and invariance under the replacement of the mapping by its inverse, because of the condition on the space of mappings considered. These are all generic requirements, which any self-matching functional must satisfy.

Invariance under rotations of $\mathbb{R}^2$ will be achieved by expressing each term as an inner or cross product between tangent vectors or elements of $\mathbb{R}^2$ itself. Invariance under rotations will follow because elements of $\mathbb{R}^2$ will only enter as differences, thereby effectively making them elements of the tangent space. Invariance under scaling will not be fully achieved; the energy will scale polynomially under scaling, which will result in a scaling of the solution as expected. The invariance requirements associated to S^1 will be achieved as follows. Coordinate invariance will be immediate as we will express the energy in a coordinate-free way, while arbitrary reparametrizations will be shown to pull back the energy density from S^1 to itself, and hence will make no difference to the energy. The same will hold true for the transformation $m \longrightarrow m^{-1}$. Before describing the details of the energy density we discuss the implications of the reparametrization invariance requirement.

[3]Actually, for the present purposes we only require the minimum argument of the energy functional to be invariant.

A reparametrization of Γ is a replacement of Γ by $\Gamma\epsilon$ where ϵ is any orientation-preserving differentiable bijection from S^1 to itself. Reparametrization invariance is immediate if the energy is expressed as a functional of $\hat{s}$ as $\mathrm{Im}(\Gamma)$ is unchanged: $\hat{E}(\hat{s}, \Gamma) = \hat{E}(\hat{s}, \Gamma\epsilon)$. It is clear from the commutative square above that this translates into the following requirement: $E(s, \Gamma\epsilon) = E(\epsilon s \epsilon^{-1}, \Gamma)$.

Each term of the energy functional will have the following form:

$$\begin{aligned} E(s,\Gamma) &= \int_{S^1} \mathcal{E}(s,\Gamma) \\ &= \int_{S^1} A(s,\Gamma) *_\Gamma A(s,\Gamma) \end{aligned} \tag{A.2}$$

where A is a one-form on S^1 and $*_\Gamma$ is the Hodge star operator associated to the metric on S^1 pulled back by Γ from $\mathbb{R}^2$. This inner product form is coordinate-free and hence guarantees coordinate invariance. If we replace Γ by $\Gamma\epsilon$ and use the relation $*_{\Gamma\epsilon} = \epsilon^* *_\Gamma \epsilon^{*-1}$, we see that provided $A(s, \Gamma\epsilon) = \epsilon^* A(\epsilon s \epsilon^{-1}, \Gamma)$, then $\mathcal{E}(s, \Gamma\epsilon) = \epsilon^* \mathcal{E}(\epsilon s \epsilon^{-1}, \Gamma)$, and the condition on E is satisfied. We will construct our energy density to satisfy this requirement.

To implement the co-circularity and stretching criteria described in section 2 we first define the following zero-forms on S^1:

$$\phi_\pm = \Gamma \pm s^*\Gamma \tag{A.3}$$

and the following products on the tangent bundle of $\mathbb{R}^2$:

$$\langle u, v \rangle = \delta_{\mu\nu} u^\mu v^\nu \tag{A.4}$$

$$\langle u, v \rangle_\times = \epsilon_{\mu\nu} u^\mu v^\nu \qquad \mu, \nu \in \{1, 2\} \tag{A.5}$$

where $\delta_{\mu\nu}$ is the 2×2 unit matrix, and $\epsilon_{\mu\nu}$ is the 2×2 anti-symmetric matrix with $1, 2$ entry equal to 1. Summation is assumed over repeated indices.

Using these we can define the terms for co-circularity and stretching.

$$\begin{aligned} \mathcal{E}_{\text{co-circ}} &= \langle \mathrm{d}\phi_-, \phi_- \rangle *_\Gamma \langle \mathrm{d}\phi_-, \phi_- \rangle + \langle \mathrm{d}\phi_+, \phi_- \rangle_\times *_\Gamma \langle \mathrm{d}\phi_+, \phi_- \rangle_\times \\ \mathcal{E}_{\text{stretch}} &= \langle \mathrm{d}\phi_-, *_\Gamma \mathrm{d}\phi_+ \rangle *_\Gamma \langle \mathrm{d}\phi_-, *_\Gamma \mathrm{d}\phi_+ \rangle \end{aligned} \tag{A.6}$$

This is the energy density for the continuous pieces of the self-matching. At each discontinuity there is a delta function density. Let the discontinuities occur at the points in the set $D = \{x_i \in S^1 : i \in I\}$, where I is some index set and we take the function to be defined at the lower parameter value at any discontinuity. Then we must add to the energy density a term

$$\mathcal{E}_{\text{discont}} = \sum_{x_i \in D} \boldsymbol{\delta}(x_i) \langle \Gamma \circ s(x_i) - \Gamma(x_i), \Gamma \circ s(x_i) - \Gamma(x_i) \rangle^{\frac{1}{2}} \tag{A.7}$$

where $\boldsymbol{\delta}(x_i)$is the one-form delta function at x_i. This calculates the perimeter of the triangle formed by the three pairs of points at any discontinuity, as described in section 2.

A simple discretization of the integral into a sum, with derivatives defined by differences, is used to compute the energy in the algorithm.

A.3. **Concept-dependent model.** Here we describe the top-down model of section 2.2.

We do not explicitly model concept space $\mathcal{C}$ except to say that we are dealing with the space of basic-level concepts. We thus feel free to use a space $\mathcal{R}_{\mathcal{C}}$ which is a representation of these concepts as shapes. The fibre above a concept c will consist of a set of contours representing the outlines of examples of that concept together with their corresponding self-matchings, defined using the energy function $E_{\mathcal{I}}$. We do not attempt to describe the exact nature of the sets of contours because in the work described here we actually fix the point in representation space that will be used (the "target" in the main text), leaving variability of the concept to future work.

To define the space of matchings between representations, $\mathcal{M}$, we need to define the notion of a morphism between two self-matchings. A morphism between two self-matchings will be a continuous map between the graphs of the self-matchings in $S^1 \times S^1/\mathbb{Z}_2$ that preserves the tree structure and the orientation of the contour. For the moment we assume that this map is a bijection, which implies a tree isomorphism. In the future it may prove advantageous to loosen this restriction. Notice that the morphisms may be divided into equivalence classes given by the tree isomorphism. We will impose a severe restriction on the morphisms included in $\mathcal{M}$, for purely computational reasons. From each class we pick the morphism that maps the edges linearly into one another. This dramatically reduces the size of the matching space $\mathcal{M}$ and makes the problem *computationally* much more efficient.

A.3.1. *Dynamics.* Because of the restriction imposed on self-matching morphisms in the previous section, our morphisms are now determined by graph isomorphisms between the tree structures. Although the continuous data has been eliminated from the specification of the morphisms, it need not be eliminated from the calculation of the energy and indeed we do not eliminate it. The edges of the tree are mapped linearly into one another, which along with the self-matching of each edge, gives a bijection between the contour segments corresponding to each edge. Again we will require the energy functional to be local, so that we will integrate an energy density along the edges to obtain the energy. Now given an isomorphism T between two self-matchings $\hat{s}$ and $\hat{s}'$, whose tree structures we will denote by $\langle V, X\rangle$ and $\langle V', X'\rangle$, we define the energy of the isomorphism to be

$$E_{\mathcal{C}}(T) \quad = \quad \sum_{x \in X} e_{\mathcal{C}}(T_x) \tag{A.8}$$

Here T_x is the continuous map between the edges x and $T(x)$ and e is the functional that computes its cost.

References

1. R. Basri, L. Costa, D. Geiger and D. Jacobs, "Determining the Similarity of Deformable Shapes", Physics Based Modeling Workshop in Computer Vision, 1995.
2. H. Blum, "Biological Shape and Visual Science", *J. of Theoretical Biology*, **38**:205-287, 1973.
3. H. Blum and R.N. Nagle, "Shape Description using Weighted Symmetric Axis Features", *Pattern Recognition*, Vol. 10, pp. 167-180, 1978.
4. C.A. Burbeck and S.M. Pizer, "Object representation by cores: Identifying and representing primitive spatial regions.", *Vision Research*, Vol. 35, 1995.
5. T.J. Cham and R. Cipolla, "Geometric Saliency of Curve Correspondences and Grouping of Symmetric Contours", *Proc. 4th European Conf. on Computer Vision*, Lecture Notes in Computer Science 1064, pp383-398, London, 1996.
6. H. Freeman, "Shape description via the use of critical points", *Pattern recognition*, 10:159-166, 1978.
7. Y. Gdalyahu and D. Weinshall, "Measures for Silhouette Resemblance and Representative Silhouettes of Curved Objects", *4th ECCV*, Cambridge, UK, 1996.
8. D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos. "Dynamic programming for detecting, tracking and matching elastic contours." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-17, March 1995.
9. T-L.Liu, D. Geiger, and R. V. Kohn, "Representation and Self-Similarity of Shapes", *International Conference in Computer Vision*, Mumbai, India, 1998.
10. Gorman, J., R. Mitchell, and F. Kuhl, 1988, "Partial Shape Recognition Using Dynamic Programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(2):257–266.
11. Huttenlocher, D., G. Klanderman, and W. Rucklidge, 1993, "Comparing Images Using the Hausdorff Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(9):850-863.
12. D. P. Huttenlocher and S. Ullman, "Object Recognition using Alignment", *Proceedings First International Conference on Computer Vision*, pp 102-111, London, 1987.
13. Kass, M., A. Witkin, and D. Terzopoulos, 1988, "Snakes: Active Contour Models," *Int. J. Comp. Vis.***1**(4):321-331.
14. F. Leymarie and M.D. Levine. "Simulating the grassfire transform using an active contour model." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14(1), January 1992.
15. D. Mumford, "Mathematical Theories of Shape: Do they model perception?", *SPIE Proceedings: Geometric Methods in Computer Vision*, Vol. 1570, 1991.
16. D. Mumford, R. Herrnstein, S. Kosslyn and W. Vaughan, "Analysis and synthesis of human and avian categorizations of 15 simple polygons", *Harvard University Dept. of Psych. preprint*, 1989.
17. R.L. Ogniewicz, *Discrete Voronoi Skeletons*. Hartung-Gorre, 1993.
18. S.M. Pizer, W.R. Oliver and S.H. Bloomberg, "Hierarchical Shape Description via the Multiresolution Symmetric Axis Transform", *IEEE PAMI*, Vol. 9, No. 4, July 1987.
19. J. Ponce, and O. D. Faugeras, "An Object Centered Hierarchical Representation for 3D Objects: The Prism Tree", CVGIP(38), No. 1, April 1987, pp. 1-28.

20. Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Bream, *Basic objects in natural categories*, Cognitive Psychology **8** (1976), 382–439.

21. K. Siddiqi and B.B. Kimia, "Parts of Visual Form: Computational Aspects", *IEEE PAMI*, Vol. 17, No. 3, pp. 239-251, March, 1995.

22. K. Siddiqi and B.B. Kimia, " A Shock Grammar for Recognition", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp 507-513, S. Francisco, 1996.

23. Tappert, C., 1982, "Cursive Script Recognition by Elastic Matching," *IBM Journal of Res. Develop.* **26**(6):765–771.

24. D. Terzopolous, A. Witkin,A. and. M Kass. Symmetry-seeking models and 3D object recovery. *Int. J. Comput. Vision.* 1, 211-221. 1987.

25. A. Tversky, "Features of similarity", *Psychological Review.* **84**, pp.327-352, 1977.

26. Ullman, S., 1989, "Aligning Pictorial Descriptions: An Approach to Object Recognition," *Cognition* **32**(3):193-254.

27. S.C. Zhu and A.Yuille, "FORMS: a Flexible Object Recognition and Modeling System", *Proceedings 5th International Conference on Computer Vision*, Boston, 1995.

FIGURE 1. Spot the hidden figure.

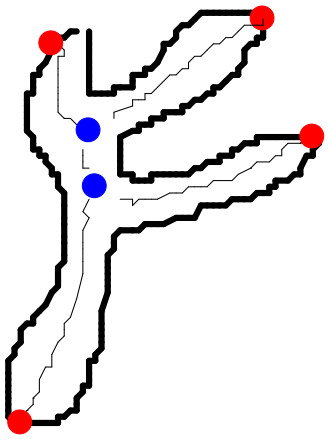


FIGURE 2. Example of self-matching represented by the median points of the line segments joining the matched points, exhibiting the degree three tree structure

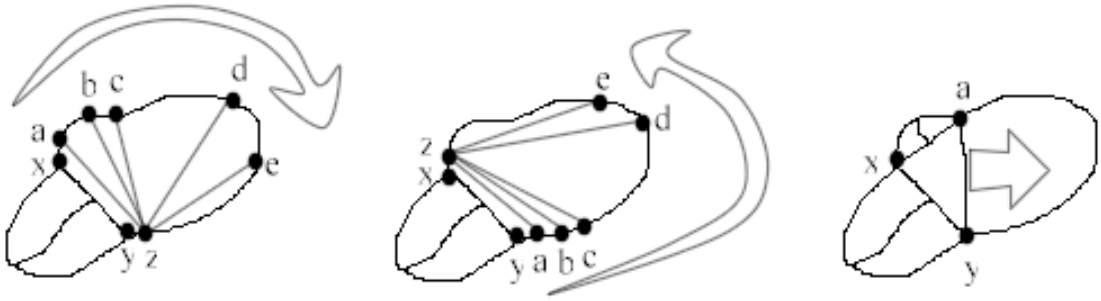


FIGURE 3. Possible moves forward in the variant of Dijkstra's algorithm used to minimize $E_{\mathcal{I}}$. The vertex removed from the queue is $\langle x, y \rangle$, and its predecessors are shown as the axis leading up to it. The point z is a neighbour of x or y in the sampled contour. The pairs $\langle a, z \rangle$, $\langle b, z \rangle$, and so on are possible next vertices in the path with increasing values of stretching energy. These are the continuous moves. The discontinuous moves are very similar except that the point z is constrained to be the same as either x or y rather than a neighbour, and the points a etc. must be at least one away from x or y. The third diagram shows how the discontinuous moves can only take place (in this case the move to $\langle a, y \rangle$ from $\langle x, y \rangle$) if there is another path with which the move can merge (in this case the path ending at $\langle x, a \rangle$).

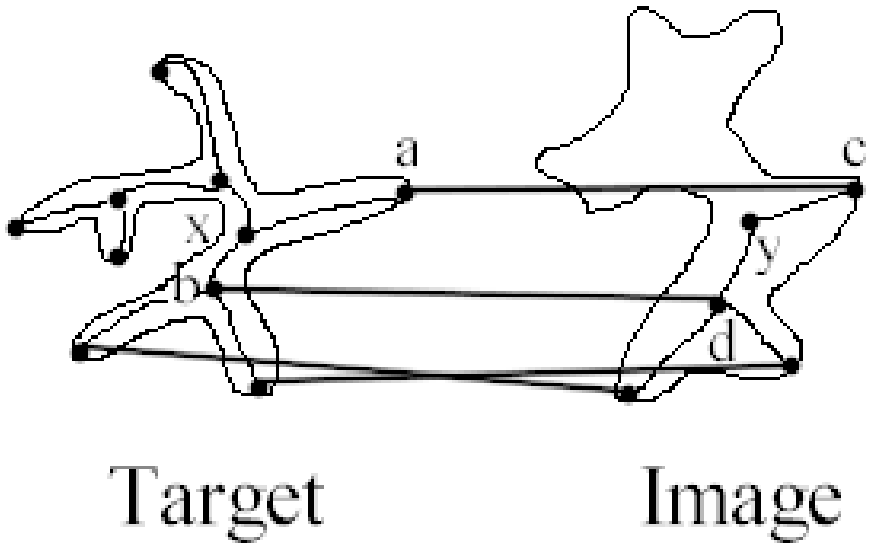
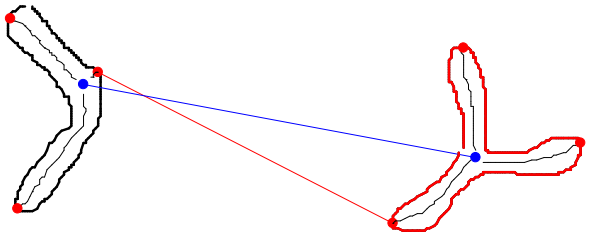


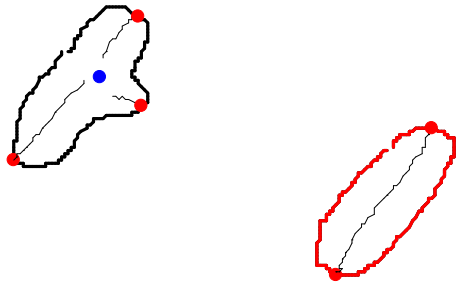
FIGURE 4. An illustration of a typical stage in the algorithm described in section 3.2. When point y is removed from the queue its boundary and discontinuity predecessors c and d have already been mapped to the target nodes b and a. This leaves only one choice of mapping for y, to whit, to x. A new vertex of the second class is created, with predecessors c and d, but with energy equal to that of y plus the energies of matching edges (c, y) to (a, x), and (d, y) to (b, x). This new vertex is placed in the queue.

(a)

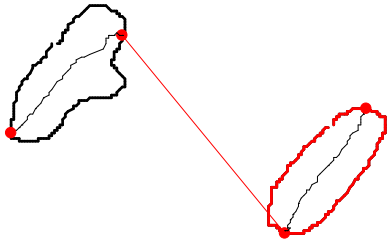


(b)

FIGURE 5. Example 1. In (a) the self-matching is computed without the target on the right. In (b) the target has been used. The lines connecting the target to the image show how the nodes are matched. We have omitted some of these lines for clarity.



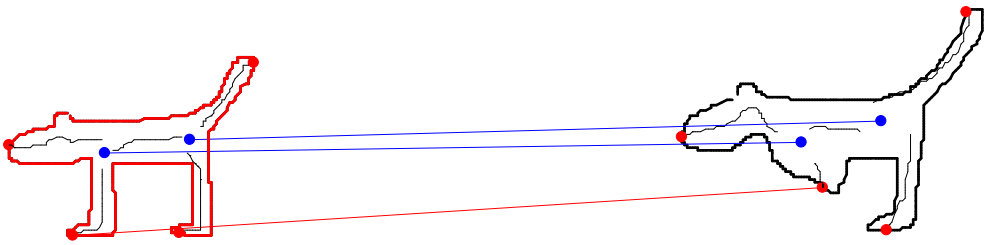
(a)



(b)

FIGURE 6. Example 2. Again, (a) is computed without the target on the right; (b) uses the target.

(a)



(b)

FIGURE 7. Example 3. The target is on the left. (a) does not use the target, (b) does.